
0. ApiExps Routine Library Introduction and Study Guide Table of Contents

| | |
|--|---|
| 1. Overview of Routine Library Functionality..... | 1 |
| 2. Study routes | 1 |
| 2.1 3D Scene Development Dependent Tools..... | 2 |
| 2.1.1 3D Modeling Tools and Required Plug-ins..... | 2 |
| 2.1.2 3D engine and required plug-ins | 2 |
| 2.2 Simulation Scene Interaction Interface | 3 |
| 3. Key Function Index..... | 1 |
| 4. Wonderful case graphic display | 3 |
| Additional resources | 3 |

1. Routine Library Features Overview

This routine library contains the configuration of the experimental platform required for the development of 3D scene modeling and simulation of the RflySim platform and the methods of calling the platform's 3D scene simulation-related functional interfaces, and these interactive interfaces in addition to the built-in console commands and shortcuts, but also includes ways to call external programs, which [\[Installation directory\]\RflySimAPIs\RflySimSDK\ue](#) under the Python interface libraries are as follows [UE4CtrlAPI.py](#), [UEMapServe.py](#)

2. Key Function Index

| serial number | point of knowledge (math.) | Implementation routes |
|-------------------|---|---|
| 1 | RflySim3D 3D scene development will use what dependent tools, including 3DMax, SketchUp, Unreal Engine and other software tools, installation and configuration tutorials and simple use of the introduction | e0_DevToolsUsage\Index.pdf |
| 2 | How to use shortcut keys for interaction in RflySim3D, including using keyboard and mouse to control the viewpoint, switching maps, and other operations | e1_KeyboardAPI\Intro.pdf |
| 3 | How to interact with console commands in RflySim3D, including using console commands to adjust scene objects, statistics on rendering performance consumption, and more. | e2_CommandAPI\Intro.pdf |
| 4 | How to set up the initialization of RflySim3D software when simulating, including loading and saving scene levels and objects in the levels, setting resolution and refresh rate, etc. | e3_InitAPI\Intro.pdf |
| 5 | How to adjust the hierarchy of the vehicle 3D model in RflySim3D, including modifying the xml configuration file to define the motion associations of different actuators in the model as well as defining the motion associations between different models through external interfaces. | e4_UAVCtrl\Intro.pdf |
| 6 | How to call the map service interface of RflySim platform, including loading map data (mainly elevation grayscale map here) and real-time solving the corresponding positional terrain height in UE (including the transformation of the map datum), in order to realize real-time terrain matching | e5_UEMapCtrl\Intro.pdf |
| 7 | How to quickly call the common scene interaction interfaces of RflySim3D, and manipulate the objects in the scene and get the data of different objects in the scene by combining batch scripts and Python interfaces. | e6_RflySim3DCtrlAPI\Intro.pdf |
| 8 | How to call the built-in GIS services in RflySim3D, including the use of the built-in Cesium large scene in RflySim3D and the method of modifying the GPS origin of the large scene | e7_RflySim3DGIS\CesiumPlugin\Readme.pdf |

| | | |
|---|---|---|
| 9 | How to use the built-in scene effects in RFLySim3D, including 3D virtual pipelines, communication effects, explosion effects, and models with special configurations. | e8_RflySim3DEffectIntro.pdf |
|---|---|---|

3. Study routes

3.1 3D Scene Development Dependent Tools

- See [e0_DevToolsUsage\Index.pdf](#) for implementation routines.

3D scene development dependent tools are software for creating and editing resources such as 3D models, textures, materials, animations, and software for organizing these resources into interactive virtual environments.

3.1.1 3D modeling tools and required plug-ins

Existing 3D engines are already compatible with most of the processes and resources of 3D scene development, but for some specific simulation needs, not every developer has the ability or need to use 3D engines to build their own 3D resources from scratch.

Therefore, some mature 3D modeling tools, such as 3DS Max, Maya, Blender, etc., can be used to create or edit high-quality 3D models. At the same time, in order to improve the reusability and compatibility of 3D resources, it is also necessary to install some plug-ins specialized in exporting or converting formats, such as FBX Exporter, GLTF Exporter, DataSmith and so on. In this way, 3D resources in different formats can be easily imported into the 3D engine for further processing.

3.1.2 Unreal Engine and required plug-ins

To develop 3D simulation scenarios using the Unreal Engine, in addition to installing the latest version of Unreal Engine 4 (UE4), you need to install a number of specialized plug-ins and extension packages, depending on the needs of different projects. For example, in order to use earth-level terrain data, Cesium for Unreal needs to be installed. In order to support the control and dynamics modeling of aircraft, the AirSim plug-in can be installed, which provides a complete set of UAV simulation framework and is compatible with the RFLySim platform. In addition, in order to enhance the rendering effect and physical performance of the Unreal Engine, you can install NVIDIA's Gameworks series of plug-ins, such as Flex, Flow, HairWorks, etc. These plug-ins can realize complex visual effects such as fluids, flames, and hairs. There are other plug-ins such as World Machine, Terrain Composer, etc. that can be used to generate and edit high-quality terrain and environments. All of these plugins can be downloaded from the official Unreal Engine website or third-party websites.

3.2 Simulation Scene Interaction Interface

Based on the Unreal Engine, RflySim3D has built-in scene elements (including 3D scenes, model resources, and interactive interfaces, etc.) for various simulation purposes, which can be freely manipulated, and the platform provides corresponding interfaces to realize such interactive functions. Firstly, there are built-in console commands and shortcuts, which can be invoked either directly in RflySim3D or through external scripts. Then the platform also provides Python interface libraries related to 3D scene interaction, including [UE4CtrlAPI.py](#), the scene interaction interface of RflySim3D, and [UEMapServe.py](#), the map server of RflySim platform, and the same functionality interfaces are also available in MATLAB version.

4. Wonderful case graphic display

See <https://rflysim.com/doc/zh/3/CoreExp.html>

Additional resources

Official Documentation: RflySim Official Documentation: <https://rflysim.com/doc/zh/>

Community Exchange: Join RFLYSim Technical Exchange Group: 951534390

